Mean field games modelling of pedestrian flows

Théophile BONNET

October 16, 2020

LPTMS-Saclay and IJClab

Under the supervision of Cécile Appert-Rolland (IJClab) and Denis Ullmo (LPTMS-Saclay)

20-04-2020

Abstract

In this work, we adapt mean field games formalism to crowd modelling. We implement the case of a crowd crossed by a hard cylinder that has been studied experimentally in [1] in order to see if we can reproduce several phenomena specific to crowd dynamics that are not modelled by agent-based or fluid dynamics. To that end, we start from a partly nonlinear code previously used by Thibault Bonnemain in [2] and compare its efficiency with a totally nonlinear code applied to the same system of equation. We also check the conservation of relevant quantities, and we extend the whole to 2-D. Special attention is given to boundary conditions and conservation of the mean field game equivalents of mass and energy. The comparison between experiments and simulation shows a good qualitative correspondence between computational and experimental result, denoting the importance of long term anticipation in crowds.

Dans ce rapport, nous adaptons le formalisme de la théorie des jeux en champs moyen à la modélisation d'une foule. Nous implémentons le cas d'un cylindre rigide qui a été étudié expérimentalement par Alexandre Nicolas et al[1] afin de tenter de reproduire plus phénomènes spécifiques à la dynamique des foules et qui ne sont pas décrits par les modèles de dynamique des fluids et basés sur des agents. Pour cela, nous partons d'un code partiellement non-linéaire précédemment utilisé par Thibault Bonnemain dans [2] et nous comparons son efficacité avec un code entièrement non-linéaire appliqué au même système. Nous vérifions aussi la conservation de certaines quantités importantes et nous étendons le tout à la 2-D. Une attention particulière est accordées aux conditions aux limites et à la conservation des équivalents en champs moyens de la masse et de l'énergie. La comparaison entre l'expérience et la simulation montre une correspondance qualitative entre les résultats expérimentaux et simulés et met en évidence l'importance de l'anticipation à long terme dans les foules.

Contents

1 Introduction

In recent years, there has been more and more communication between very different fields, often yielding new directions for studying preexisting problems. Especially, physicists have started to be interested in subjects they have neglected until now, and by doing so they try to solve them or to get an insight of their solutions by using a physicist mindset. Crowd dynamics is one these subjects. Indeed, for a long time crowds were modeled by fluid dynamics or simple agent-based models, yielding some good results while exhibiting significant weaknesses. Most of recent works studied the low density limit and tried to include the micro scale to the models under different forms. For instance D.Helbing & al [3] conceptualized the idea of social pressure under the form of a **social force**, by analogy with physical forces, and while it provides interesting results, the lack of anticipation will be shown to be repulsive later on. On the other hand, Van den Berg & al [3]tried to include anticipation into the model but only managed to do so on very short time-scale - the pedestrians basically avoid collision for the next few seconds by choosing their speed while assuming the speed of others will remain constant - which is not enough in some cases. Moreover, we have to note that we see the same kind of behaviors, irrelevant from density (at least in the low density case).Lastly, the general idea of the community was that if we were to go to the high density limit, the problem would reduce to a purely mechanical problem well described by granular dynamics[4]. This was shown experimentally false by Alexandre Nicolas & co [1]. This experiment consisted in a hard cylinder crossing a dense crowd of static people, as shown on fig 1, yielding the density field on fig 2 in the frame of the cylinder.

Fig 2 exhibits a highly non-trivial behavior. Indeed, we see a depletion of density in front and behind the cylinder while we have excess density on the side. This means that pedestrians may accept a temporary high density if it allows them to be more at ease later on, and this shows existence of a long-term anticipation that was not described by previous models. Moreover, comparison between fig 3-4 shows that the velocity fields for pedestrians definitively don't behave as for granular media, even at high density. This calls for a new model including both physical forces **and** long term anticipation. A credible tool to treat this problem has been devised by mathematicians. They used the concept of mean fields we have in Physics to construct a theoretical framework allowing studies of problems where a great number of agents tries to optimize a function while taking into account the strategies of other agents. This is called mean field games. Now, if we think about crowds, it seems quite evident that members of these crowds do not behave in a strictly mechanical way: people will try to optimize their comfort while in the crowd, and they do so while trying to predict what will happen in a certain distance around them. A balance between current comfort or discomfort and future gain is then established. If they see a bottleneck, they will try to find alternative solution in advance, and if they find a cylinder going their way, they will try to avoid it preferably before it gets close. The velocity field obtained in fig 4 can be intuitively understood by taking into account that people know that if they just move back in front of the cylinder, it will still come

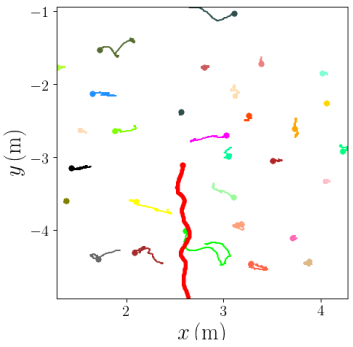

Figure 1: Experimental setup in [1]

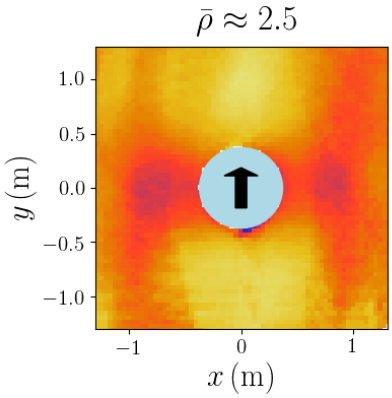

Figure 2: Density field in the frame of the cylinder in [1]

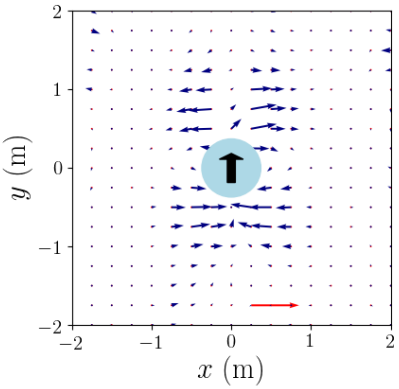

Figure 3: Experimental velocity field obtained by Alexandre Nicolas & al[1]

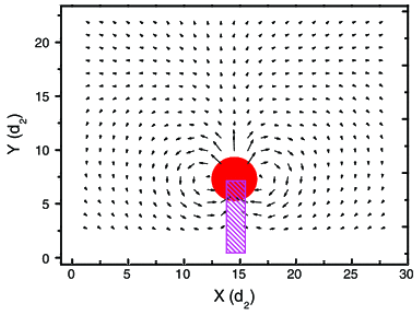

Figure 4: Velocity field in 2d dense granular layer obtained by Kolb & al[4]

to them: going on the side is then a balance between the discomfort they will have because of high density, and the future comfort they expect after that. It seems henceforth quite sensible to try and model this crowd by using mean-field games. Hence, in a first part we will introduce mean-field game formalism and justify a choice of interaction between agents. With a consideration on efficiency, and because we expect the 2d code to be very time-consuming, we will then compare two 1d algorithms in order to choose the one we will implement in 2d. We then go into the heart of the subject and dwell into 2D simulation of our mean-field game system, and we will conclude by comparing our 2D computational results with experiments, showing a good qualitative agreement between them. On the whole, this work comes as a follow-up of the thesis by Thibault Bonnemain. I did this whole internship in confinement with the very regular supervision of Denis Ullmo from LPTMS-Saclay, Cécile Appert-Rolland from IJClab ,and Thibault Bonnemain who kindly accepted to help.

2 Mean-Field Games

2.1 Some elements of game theory

Before defining mean-field game, we need to introduce some elements of game theory that are needed to get at least a feel of what we will be working with. Mathematicians and sociologists have developed game theory as an attempt to formalize the concept of strategies in games and apply it to other situations. So by "game", we actually denote any situation where any number of players (or agents) can choose from any number of strategies while trying to maximize its utility function. This utility function can represent anything, from money to comfort. A strategy is a set of actions the agent may realize depending on what he knows about the other players. This simple definition already suggests four hypothesis we have to fix to restrict the scope of the games we will consider.

- Cooperation: There is no external force enforcing cooperation between agents, but they may naturally like to stick together, or to get away from one another. This may take the form of an interaction.
- Rationality: Agents are purely rational and egoistical. They only want to optimize their utility function.
- Adaptation: Agents may adapt their strategies as the game progresses and they gain information.
- Information: Agents know everything that happened until current time, and they can predict what will happen next, up to a noise that may play a significant role.

This set of hypothesis serves as a basis to discuss general solutions of a game. Solutions in game theory differ from solutions in physics, as it denotes the strategies leading to a given final state but not the final state itself. This leads to the introduction of what is called a "solution concept" defining what is a solution in its game theory sense. The most commonly used solution concept is the Nash equilibrium. It is defined as a strategy vector (one strategy for each player) from which none of the players can stray away without losing something. Hence, in the following we will assume that there always exists a Nash equilibrium and that we go naturally to that equilibrium. This is equivalent to the pure rationality hypothesis we have made just before.

2.2 Hamilton-Jacobi-Bellman equation (HJB)

Because of our previous hypothesis, we know we are **not** deterministic and that we have uncertainties arising from the set of strategies. This can be modelled by an usual Langevin dynamics

$$d\vec{X}_t = \vec{a}_t dt + \sigma d\vec{W}_t \tag{1}$$

where $\vec{a}$ is the control parameter, σ a constant and $\vec{W}$ a Gaussian white noise of variance 1. We **choose** the following cost function to be minimized

$$c[\vec{a}](\vec{X}, t) = \mathbb{E}\left[\int_t^T \left(\frac{\mu}{2}\vec{a}_\tau^2 - V(\vec{X}_\tau, \tau)\right) d\tau + c_T(\vec{X}_T)\right] \tag{2}$$

with T the final time, V an exterior potential or *environmental gain*, μ a constant and c_T a *terminal cost* at which we want to process to find itself at the end. The square dependency in $\vec{a}$ restricts the study to quadratic games. We chose this for two main reasons: to simplify the study while maintaining enough complexity to keep it interesting, and because of the possible transformation toward nonlinear Schrödinger equation that will become apparent later on. On the other hand, by environmental cost we mean that we suppose there can be some sort of exterior force applied onto the agents, not enforcing cooperation but making space more or less attractive for them. We then introduce the value function

$$u(\vec{X}_t, t) = \inf_{\vec{a}} c[\vec{a}](\vec{X}, t) \tag{3}$$

and, using dynamic programming with Bellman's optimality [5] principle, Ito's lemma [6] and after optimization, we get the *Hamilton-Jacobi-Bellman equation*

$$\begin{cases} \partial_t u + \frac{\sigma^2}{2}\Delta u - \frac{1}{2\mu}||\vec{\nabla} u||^2 &= V \\ u(\vec{X}, T) = c_T(\vec{X}) \end{cases} \tag{4}$$

This equation constitutes a *noisy* alternative to Euler-Lagrange equation and describes the dynamics of an optimization with quadratic running cost for one agent. If we were to take an analogy with physics, the usual physical action would be an utility functional we want to optimize, the Lagrangian would be a running cost and the usual $\vec{q}$ is a control parameter. We will make use of this later on.

Now, in mean-field games we are interested in what happens for a great number of interacting agents. As a first step we introduce the notion of differential games, which is just a generalization of HJB equation to a multi-agents case. The state of the player i is given by $\vec{X}_t^i$ and its strategy is adjusted in real time. The cost function each agent then tries to optimize now becomes a cost functional

$$c[\vec{a}^1, ..., \vec{a}^N](\vec{X}^1, ..., \vec{X}^N, t) = \mathbb{E}\left[\int_t^T \left(\frac{\mu^i}{2}(\vec{a}_\tau^i)^2 - V^i(\vec{X}_\tau^1, ..., \vec{X}_\tau^N, \tau)\right) d\tau + c_T^i(\vec{X}_T^1, ..., \vec{X}_T^N)\right] \tag{5}$$

and by defining the usual value function, we get the HJB equation for a N-players game

$$\begin{cases} \partial_t u^i + \frac{\sigma_i^2}{2} \sum_{j=1}^N \Delta_{x^j} u^i - \sum_{\vec{i \neq j}} \frac{1}{\mu^j} (\vec{\nabla}_{x^j} u^j).(\vec{\nabla}_{x^j} u^i) - \frac{1}{2\mu^i} ||\vec{\nabla}_{x^j} u^i||^2 = V \\ u_T^i(X_T^1, ..., \vec{X_T^N} = c_T^i(X_T^1, ..., X_T^N) \end{cases} \tag{6}$$

This set of equations henceforth describes the behavior of N-players adjusting their strategies in time and whose state vectors has some uncertainties described by a Langevin dynamics

$$d\vec{X}_t^i = \vec{a}_t^i dt + \sigma_i d\vec{W}_t^i \tag{7}$$

These players are purely rational and nothing forces them to cooperate so they will try to maximize their own cost functional without regards for others. We now have taken into account the four assumptions we started with in our part on game theory, and we are left with a set of equations that becomes intractable as soon as N becomes big. We finally have to introduce the mean field part of the theory for the whole to become usable.

2.3 Mean field games

The deterministic mean-field games formalism we are going to use has been introduced by P-L.Lions and J-M.Lasry [7]-[8]. It is obtained by taking the $N \to \infty$ limit. In that formalism, the potential V^i is supposed to depend on the other players behavior only through the empirical density

$$\tilde{m}(x,t) = \frac{1}{N} \sum_{j=1}^N \delta(x - X_t^i) \tag{8}$$

hence yielding $V^i = V[m](X_t^i)$ and $c_T^i = c_T[m](X_T^i)$. Finally we average this empirical density over all the realisation of the noise and get the mean field $m(x,t) =< \tilde{m}(x,t) >$. Defining the corresponding value function as we did previously, we can reduce the multi-agents system to a one body system verifying HJB equation, with decoupled agents

$$\begin{cases} \partial_t u + \frac{\sigma^2}{2} \Delta u - \frac{1}{2\mu} ||\vec{\nabla} u||^2 = & V[m] \\ u(x,T) = & c_T[m](x) \end{cases} \tag{9}$$

To ensure consistency we need another equation for m. This is obtained by using the fact that agents follow a Langevin dynamics. Hence, for a large enough number of agents we get a Fokker-Planck equation arising, and the final system reads

$$\begin{cases} \partial_t u + \frac{\sigma^2}{2}\Delta u - \frac{1}{2\mu}||\vec{\nabla} u||^2 & = V[m] \\ u(\vec{x}, T) & = c_T[m](\vec{x}) \\ \partial_t m - \frac{1}{\mu}\vec{\nabla}.[m\vec{\nabla} u] - \frac{\sigma^2}{2}\Delta m & = 0 \\ m(\vec{x}, t=0) & = m_0(\vec{x}) \end{cases} \tag{10}$$

With that, we have our full description of quadratic mean-field games in its deterministic formulation. It is important to note that this is not a classical forward system equation: while HJB equation is backward in time, meaning we know the final state and deduce the trajectory from that, Fokker-Planck equation is forward in time, as we know the initial time and see how it evolves. This coupling makes treatment of this system highly non trivial, even though we can find some alternative representation as discussed in the next subsection.

2.4 Nonlinear Schrödinger formulation (NLS)

Indeed, one can obtain a backward NLS equation by applying *Cole-Hopf transformation* on HJB equation[9]. Similarly, by doing an hermitization of FP equation

$$\begin{cases} \phi & = e^{-\frac{u}{\mu\sigma^2}} \\ \Gamma & = m e^{\frac{u}{\mu\sigma^2}} \end{cases} \tag{11}$$

one then obtain a system of forward-backward NLS equation

$$\begin{cases} \mu\sigma^2 \partial_t \phi & = -\frac{\mu\sigma^4}{2}\Delta\phi - V[m]\phi \\ \mu\sigma^2 \partial_t \Gamma & = \frac{\mu\sigma^4}{2}\Delta\Gamma + V[m]\Gamma \end{cases} \tag{12}$$

with initial and terminal conditions

$$\begin{cases} \phi(x, t=T) & = f(x) \\ \Gamma(x, t=0) & = \frac{m_i(x)}{\phi(x, t=0)} \end{cases} \tag{13}$$

One must note several peculiarities of this system. First, the forward-backward structure is retained. Second, contrary to the usual NLS equation, we have no imaginary part in the equations. This means in particular that ϕ and Γ are **not** a kind of probability complex amplitude: they are non-periodic, positive functions. One must also take into account the fact that ϕ and Γ are **not** conserved while their product has to be conserved, as HJB and FP equations conserve mass. Nonetheless, we will use this formulation for two reasons: first, it allows for a simpler implementation in the simulation code. One can just code one solver and apply it to $\tilde{\phi}$ the time reversed ϕ, and Γ. Second, there is a whole bunch of tools constructed by physicists for usual NLS equation over the years that can easily be adjusted to this system. Lastly, one can write an action for this formulation

$$S[\Gamma,\phi]=\int_0^T dt\int_{\mathbb{R}} dx\left[\frac{\mu\sigma^2}{2}(\Gamma\partial_t\phi-\phi\partial_t\Gamma)-\frac{\mu\sigma^4}{2}\vec{\nabla}\phi.\vec{\nabla}\Gamma+U[m]\right] \tag{14}$$

where $U[m]$ is the functional anti-derivative of $V[m]$. This, along with time translation invariance, implies by way of Noether theorem that there exists a conserved quantity that we will call *energy*

$$E=\int_{\mathbb{R}} dx\left[\frac{\mu\sigma^4}{2}\vec{\nabla}\phi.\vec{\nabla}\Gamma+U[m]\right] \tag{15}$$

by analogy with physics. Actually, its nature will depend on the studied problem, but this conservation, along with mass conservation, provides for a good way to check our results. For clarity sake we will refer to the first term of the energy as the *kinetic energy* of the game, as it is related to the gradients of $p\phi$ and Γ, hence to the speed of the agents, and we will call the second term the *potential energy*. Lastly, for the sake of future interpretation, we need to note the existence of another alternative representation, the hydrodynamic representation, that yields the following system:

$$\begin{cases} \partial_t m+\vec{\nabla}.(m\vec{v}) & =0 \\ \partial_t\vec{v}+\vec{\nabla}\left[\frac{\sigma^4}{2\sqrt{m}}\Delta\sqrt{m}+\frac{||\vec{v}||^2}{2}+\frac{V[m]}{\mu}\right] & =0 \end{cases} \tag{16}$$

with v the "velocity" of the system

$$\vec{v}=\frac{\phi\vec{\nabla}\Gamma-\Gamma\vec{\nabla}\phi}{2m} \tag{17}$$

Although this velocity is more abstract than the usual one, it still describes the flow of mass in our distribution. If compared to an experimental speed, it should roughly be equivalent to an usual time-averaged and ensemble averaged velocity. This means that if it is 0, the crowd is still. We will now conclude this theoretical introduction by choosing the potential to be

$$V[m](x,t)=gm+U_0(x,t) \tag{18}$$

with g a negative constant and U_0 an exterior potential depending only on space and time. This corresponds to the simplest game where agents locally interact and don't like to stick together. It is to some extent quite representative of what happens in crowds, at least in a qualitative way. This choice leads to the definition of one typical length, the healing length of the system

$$\nu=\frac{\mu\sigma^4}{|g|} \tag{19}$$

that represents the typical length on which m is affected by a perturbation. On the other hand, the exterior potential will be used to models exterior forces applied to the crowd, for

instance the cylinder crossing the crowd will be modelled by a circular infinite potential wall moving at a given velocity in the simulation space.

3 Comparison between two algorithms

Before dwelling into the 2D code, it was important to check the stability, the relevance and the efficiency of the 1D algorithm, and hopefully to find a faster algorithm. Indeed, the initial algorithm used by Thibault to code the first of the two programs we studied, involved successive solving of the linear problem by means of Crank-Nicholson algorithm for NLS equations. This algorithm is unconditionally stable for normal NLS equations, but because of our forward-backward structure, we had to use "brakes" in order to achieve convergence. Each iteration time-consumption was satisfyingly small, but the big number of iterations made it so it was doubtful this was the most efficient we could find. Hence, during the thesis presentation of Thibault Bonnemain, a totally nonlinear algorithm that exhibited systematic convergence without the need for brakes was suggested [10]. In the end, the first objective of this whole work was to check the efficiency of this algorithm. For this whole part, we will denote by "Gueant algorithm" the totally nonlinear algorithm, while we will denote by "Thibault algorithm" the linear algorithm[11]. Lastly, in both case the actual matrix inversion is computed using Thomas algorithm, a sub-routine of Gauss inversion that is especially efficient for tri-diagonal matrices as the ones we will have in the 1D case. Note this last algorithm needs the matrix be diagonal dominant to be stable.

3.1 Thibault Algorithm

This algorithm had already been implemented by Thibault Bonnemain in the timespan of his thesis when I started my work. Its main idea is that starting from system (12), we follow these steps:

- Implicit discretization of the system
- Solve the linear system on ϕ with $V[m]$ constant
- Solve the linear system on Γ with $V[m]$ constant
- Compute the new value of m
- Redo until convergence of m

By taking $V[m]$ constant during the solving process we actually solve a system of linear equations using Thomas algorithm, and we converge iteratively toward the nonlinear solution of

the complete system. The good point is that because the actual solving is done on a linear system, it is quite fast. The bad part is that to achieve convergence, we take

$$m_{new} = \alpha m_{current} + (1 - \alpha) m_{old}$$

where m_{new} will be used to compute the new $V[m]$, m_{old} is what we used to compute $V[m]$ at the previous iteration, while $m_{current}$ is the raw product $m_{current} = \phi^{n+1}\Gamma^{n+1}$. We will not enter into the specifics of this algorithm, as it is not the focus of this work, but we need to remember that Crank-Nicholson algorithm for usual NLS equation is conservative, and we expect the scheme to keep this property when applied to our transformed mean-field game equations.

3.2 Guéant Algorithm

The second algorithm follows a very different recipe. As described by Olivier Guéant[10], it follows the following steps:

- Implicit discretization of the system
- Solve the nonlinear equation on ϕ with Γ constant
- Solve the nonlinear equation on Γ with ϕ constant
- Redo until m converges

Because of the fact that we solve the **complete nonlinear equation** at each solving step, the computation time of each iteration is longer than previously and we need a more complex solver (in our case we used Newton method), but the scheme ensures systematic convergence of m by means of the monotony of the sequences of ϕ and Γ without the need for "brakes". Moreover, because it is totally nonlinear, there is no explicit computation of m. To go into the specifics, from Eq. (12) we get

$$\begin{cases} \partial_t \phi^{n+\frac{1}{2}} + \frac{\sigma^2}{2}\Delta\phi^{n+\frac{1}{2}} &= -\frac{1}{\sigma^2}(\phi^{n+\frac{1}{2}})^2\Gamma^n \\ \partial_t \Gamma^{n+1} - \frac{\sigma^2}{2}\Delta\Gamma^{n+1} &= +\frac{1}{\sigma^2}(\Gamma^{n+1})^2\phi^{n+\frac{1}{2}} \end{cases} \tag{20}$$

For simplicity sake we take a simple discretization, knowing that for the numerical results we adopted the more precise discretization Thibault Bonnemain used. Hence, for the time derivative we take

$$\partial_t \phi = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta t}$$

and for the 1st order space derivative we take $\partial_x \phi = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta x}$ which yields

$$\partial^2_{xx}\phi = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\Delta x^2}$$

The equation on ϕ being a backward equation, the terms in i are implicit while the terms in $i+1$ are known. Conversely, in order to keep the implicit structure for the equation on Γ we keep the same time derivative but adopt the following space discretization.

$$\partial^2_{xx}\Gamma = \frac{\Gamma_{i+1,j+1} - 2\Gamma_{i+1,j} + \Gamma_{i+1,j-1}}{\Delta x^2}$$

These choices of discretization yield the following system of non-linear equations on ϕ and Γ:

$$\left\{ \begin{array}{ll} \frac{\phi_{i+1,j}^{n+\frac{1}{2}} - \phi_{i,j}^{n+\frac{1}{2}}}{\Delta t} + \frac{\sigma^2}{2}\frac{\phi_{i,j+1}^{n+\frac{1}{2}} - 2\phi_{i,j}^{n+\frac{1}{2}} + \phi_{i,j-1}^{n+\frac{1}{2}}}{(\Delta x)^2} & = -\frac{1}{\sigma^2} g(\phi_{i,j}^{n+\frac{1}{2}})^2 \Gamma_{i,j}^n \\ \frac{\Gamma_{i+1,j}^{n+1} - \Gamma_{i,j}^{n+1}}{\Delta t} - \frac{\sigma^2}{2}\frac{\Gamma_{i+1,j+1}^{n+1} - 2\Gamma_{i+1,j}^{n+1} + \Gamma_{i+1,j-1}^{n+1}}{(\Delta x)^2} & = \frac{1}{\sigma^2} g(\Gamma_{i+1,j}^{n+1})^2 \phi_{i+1,j}^{n+\frac{1}{2}} \end{array} \right. \tag{21}$$

As showed in [10], with this scheme and by initializing Γ at 0, for a potential $V[m]$ unchanging in sign, the sequence on n of Γ is monotonously increasing and bounded by a sup while the sequence of ϕ is monotonously decreasing and bounded by an inf. This is what ensures global convergence of the scheme. We then note that the two equations are of the exact same form by taking $\Delta t \to -\Delta t$. Hence, in the code, we will need only one solver for both. We took the forward equation for simplicity sake. In the following, we will only consider the latter and suppose there is no particular problem with the backward equation.

Now, we want to solve this equation with Newton Method. We denote by α the index on the Newton iteration for a given time step, and by n the global iteration on the whole system. Actually, we know $\Gamma_{i,\bullet}^{n+1}$ before we start solving for $\Gamma_{i+1,\bullet}^{n+1}$. As a consequence, the loop on α applies only to $i+1$.

First, introducing $\tilde{\sigma}^2 = (\sigma^2 \Delta t)/(2(\Delta x)^2)$ and $\tilde{g} = (g\Delta t)/\sigma^2$, Eq. (21) reads

$$\mathbf{E}(\Gamma_{i,\bullet}^{n+1}, \Gamma_{i+1,\bullet}^{n+1}) = 0 \,, \tag{22}$$

where $\mathbf{E} = (E_j)$ is the spatial vector with entries

$$E_j(X_\bullet, Y_\bullet) \equiv (Y_j - X_j) - \tilde{\sigma}^2[Y_{j+1} - 2Y_j + Y_{j-1}] - \tilde{g}\phi_{i+1,j}^{n+\frac{1}{2}}(Y_j)^2 \,. \tag{23}$$

Solving Eq. (22) with Newton then amounts to introduce a series of spatial vectors $\Gamma_{i+1,\bullet}^{n+1,\alpha}$ with $\Gamma_{i+1,\bullet}^{n+1,0} = \Gamma_{i+1,\bullet}^{n}$ and $\lim_{\alpha\to\infty} \Gamma_{i+1,\bullet}^{n+1,\alpha} = \Gamma_{i+1,\bullet}^{n+1}$. Writing

$$\Gamma_{i+1,\bullet}^{n+1,\alpha+1} = \Gamma_{i+1,\bullet}^{n+1,\alpha} + \delta\Gamma_\bullet \tag{24}$$

the recursion between α and $\alpha+1$ is given by

$$\mathbf{E}(\Gamma_{i,\bullet}^{n+1}, \Gamma_{i+1,\bullet}^{n+1,\alpha}) + \frac{\delta \mathbf{E}}{\delta Y_\bullet}(\Gamma_{i+1,\bullet}^{n+1,\alpha}) \cdot \delta\Gamma_\bullet = 0$$

which reads

$$\begin{aligned} E_j(\Gamma_{i+1,\bullet}^{n+1,\alpha}, \Gamma_{i,\bullet}^{n+1}) &= \left[-1 + 2\tilde{g}\Gamma_{i+1,j}^{n+1,\alpha}\Phi_{i+1,j}^{n+\frac{1}{2}}\right]\delta\Gamma_j + \tilde{\sigma}^2[\delta\Gamma_{j+1} - 2\delta\Gamma_j + \delta\Gamma_{j-1}] \\ &= M_{jj'}\delta\Gamma_{j'} \end{aligned} \tag{25}$$

where the matrix $M_{jj'}$ given by

$$M_{jj'} = \left[-1 + 2\tilde{g}\Gamma_{i+1,j}^{n+1,\alpha}\Phi_{i+1,j}^{n+\frac{1}{2}}\right]\delta_{j,j'} + \tilde{\sigma}^2(\delta_{j,j'+1} - 2\delta_{j,j'} + \delta_{j,j'-1}) \,. \tag{26}$$

This at least has the correct structure since if $\mathbf{E}(\Gamma_{i+1,\bullet}^{n+1,\alpha}, \Gamma_{i,\bullet}^{n+1}) = 0$, the $\delta\Gamma = 0$. We obtain a linear equation on $\delta\Gamma$ (resp $\delta\phi$) that we solve by matrix inversion using Thomas Algorithm (a subclass of Gaussian inversion). One can look at the appendix to see a brief extension of this development to the more complex discretization we used in the numerical computation. Finally, the code itself, coded in C++ object oriented with the Eigen library for linear algebra[12], is not to be discussed in this report.

3.3 Result comparison

In this subsection, we are interested in the comparison between the two algorithms and their convergence in the case of a simple example. we take $T = 1$, $\sigma = 0.8$, $L = 1$, $dx = \frac{1}{50}$, $dt = \frac{1}{250}$, with Neumann domain boundary conditions and the following initial and terminal conditions:

$$\begin{cases} \phi(x, t = T) = x^2(1-x)^2 \\ \Gamma(x, t = 0) = \dfrac{m_i(x)}{\phi(x, t = 0)} \end{cases} \tag{27}$$

We furthermore assume a potential of the form $V = gm$ where $g = -2$ and m is the player density. Fig 5-7 corresponds to Guéant algorithm and fig 8-10 to Thibault algorithm.

In both case, the black curve is the converged solution, and we get the same one in each case. On the other hand, the ways we converge to the solution are very different in these two cases. We see that Guéant algorithm follow the theorems established in [10] and have a very smooth but slow convergence, while Thibault algorithm pays the lack of theorem by a messy though fast convergence. Indeed, where we need 42 iterations for Guéant algorithm to converge, we only need 21 with Thibault algorithm (by optimizing the choice of α the "brakes"). To conclude on the 1D case, we compared the results obtained by Guéant algorithm with those we had by Thibault algorithm, and the two perfectly concurs. On the other hand, with Guéant algorithm,

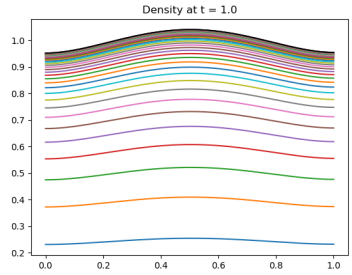

Figure 5: m at $t = T$ for Guéant algorithm

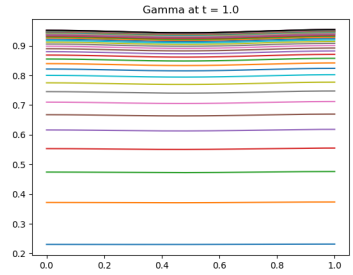

Figure 6: Γ at $t = T$ for Guéant algorithm

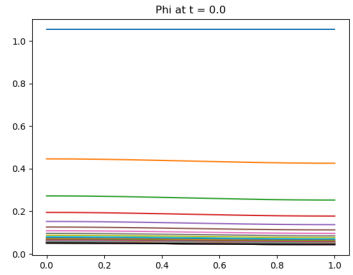

Figure 7: ϕ at $t = 0$ for Guéant algorithm

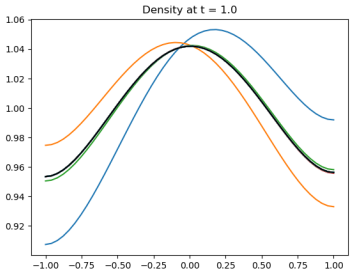

Figure 8: m at $t = T$ for Thibault algorithm

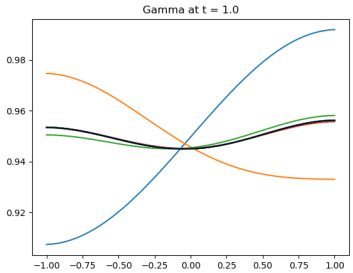

Figure 9: Γ at $t = T$ for Thibault algorithm

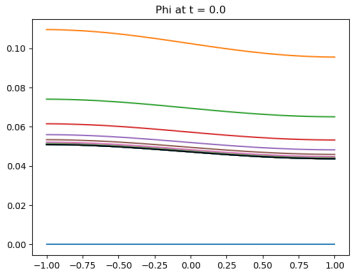

Figure 10: ϕ at $t = 0$ for Thibault algorithm

although we indeed don't need to implement brakes, both the time taken for the global loop to converge and the time taken by each iteration are too long for this algorithm to be usable in 2D. Hence we decided to stick to Thibault algorithm and to keep Guéant algorithm in case Thibault algorithm eventually fails. Lastly, we have to note that in case of a discretization too refined the matrices are no longer diagonally dominants, yielding instabilities when using Thomas inversion algorithm. This however was not an issue as we managed to keep it under control in the 1D case.

3.4 Boundary conditions and conserved observable

Before going into 2D, we will discuss a little bit about the effect of the initial and final conditions on the numerical computation. Indeed, in our first section we noted the theoretical existence of (at least) two conserved quantities we called mass and energy. However, we note some peculiarities on figure 15. The computation was done with a size $L = 5$, $T = 5$, $dx = \frac{1}{10}$, $dt = \frac{1}{200}$ and Dirichlet domain boundaries. We used the usual $V[m] = gm$ and $\sigma = 0.45$. The initial and final condition are as follows

$$\begin{cases} m_i(x) &= -\alpha x^2 + \beta \\ u_T(x) &= 0 \end{cases} \tag{28}$$

with $\beta = \frac{3}{1.6}$ and $\alpha = \frac{16}{3}\beta^3$. This corresponds to a quite narrow parabolic initial distribution and a situation where agents don't favor any space position at the end of the game. Moreover, the Dirichlet domain boundary condition is

$$\begin{cases} \phi(-L,t) = & \phi(L,t) = 0 \\ \Gamma(-L,t) = & \Gamma(L,t) = 0 \end{cases} \tag{29}$$

which translate into

$$\begin{cases} m(-L,t) = & m(L,t) = 0 \\ u(-L,t) = & u(L,t) = +\infty \end{cases} \tag{30}$$

From a "physical" point of view, the condition on u is equivalent to an infinite potential wall at the boundaries, ensuring no mass is lost. In these conditions we see on fig 14 that although the normalized mass is conserved up to 0.22% of its initial value, the energy (cf fig 15) shows a steep decrease before its stabilisation. On the other hand, we see on fig 16 that the kinetic energy goes to zero with time and that the potential energy stabilizes itself at around $V = -0.07$ which seems quite logical. Indeed, we expect the system to go to a stationary state [13] for which the distribution is constant and the speeds are null while the interaction between agents has no reason to disappear (because of conservative finite domain boundary condition). As seen on fig 17, this initial loss is due to discretization effect when the initial or final condition enforces fast dynamics, and **does not** necessarily appear clearly on the curves of m, ϕ, and Γ.

4 Into the 2D world

Now, we need to start applying this whole machinery to our crowd problem. To do so, we first implement Thibault algorithm in 2d and check qualitatively the results by analogy with the 1d case. We then implement the crossing cylinder and finally we discuss the results we got and compare them with experiments. The mean field equations in 2d reads

$$\begin{cases} \partial_t \phi^n + \frac{\sigma^2}{2}(\partial^2_{xx}\phi^n + \partial^2_{yy}\phi^n) &= -\frac{g}{\sigma^2} m^{n-1}\phi^n \\ \partial_t \Gamma^n - \frac{\sigma^2}{2}(\partial^2_{xx}\Gamma^n + \partial^2_{yy}\Gamma^n) &= +\frac{g}{\sigma^2} m^{n-1}\Gamma^n \end{cases} \tag{31}$$

On a fundamental standpoint, there is nothing really subtle in this implementation: it is essentially the same as in 1D, but there are some technical subtleties that we will discuss very briefly. Indeed, in 1D the matrices we had to invert were tri-diagonal - hence the use of Thomas algorithm - whereas they are now sparse matrices. Moreover, because of the fact that our modified NLS equations have no imaginary part, these matrices are not self-adjoint. This

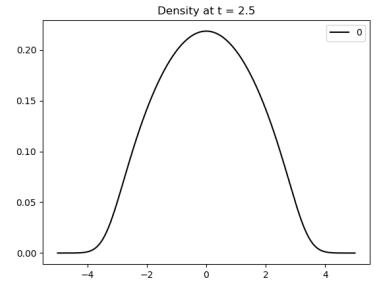

Figure 11: m at $t = T/2$

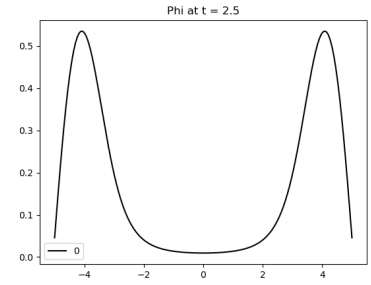

Figure 12: ϕ at $t = T/2$

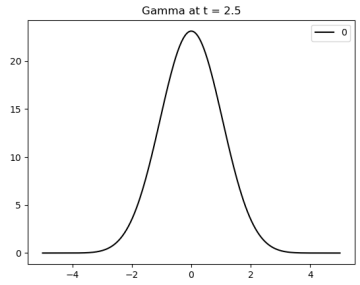

Figure 13: Γ at $t = T/2$

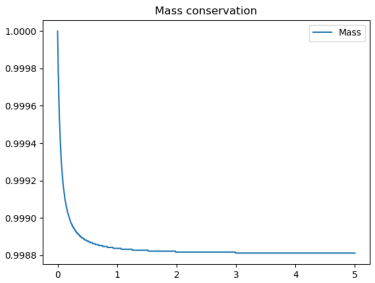

Figure 14: Mass conservation

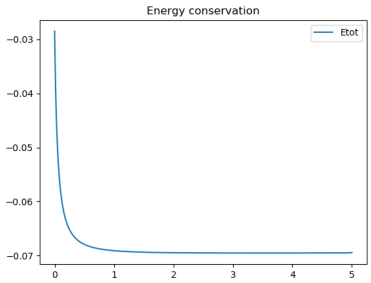

Figure 15: Total energy

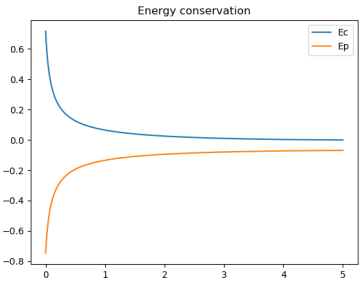

Figure 16: Kinetic and potential energy

means that we had to use a LU decomposition sparse solver included in the Eigen library. On a side note, we no longer had stability issues when increasing the precision of the discretization. The one thing that was more subtle concerned the way to model the cylinder crossing the simulation box. Indeed, whereas the discussion on the 1d algorithm was mostly there to clarify our problem, our 2d code is aiming to model real world situations. Hence it was necessary to model not only the free behavior of the system but also its reaction to external stimuli. The case of a crossing cylinder was chosen for comparison with the experimental work from Alexandre Nicolas & al [1] discussed in the introduction. The cylinder here is considered as perfectly rigid and is uninfluenced by the crowd that start by being uniformly distributed. Two ways were initially considered. First was to implement the cylinder as a moving hole in the simulation box and enforcing 0-Dirichlet domain boundary condition there. Although this had the merit of ensuring a perfect rigidity for the cylinder, it was both theoretical and technically more complex than the second way. This second way is actually just to model the cylinder by a time-dependent potential wall, high enough to ensure almost no penetration of the distribution into the wall. While this does not ensure perfect rigidity, by taking high enough potential it is a good enough approximation, and it has the merit of being both numerically and theoretically simple.

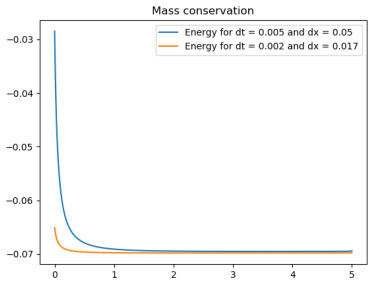

Figure 17: Energy for different discretization step width

4.1 Coherence with the 1d case

First, to check the credibility of the 2d results, we used similar parameters as the ones used for the 1d case but extended to two dimension. For instance, fig 18-20 are obtained for a 2d parabolic initial distribution, a flat final cost and Dirichlet domain boundary conditions. By doing so, we get something reminding us of fig 11-13. Mass and energy show similar behavior as in 1d, and all of this suggests there are no deep subtleties when going from the 1d case to 2d.

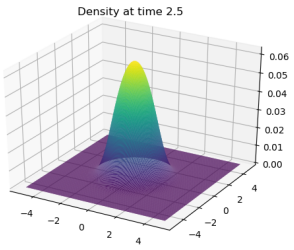

Figure 18: m at $t = T/2$

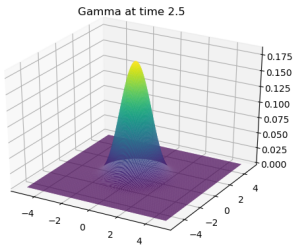

Figure 19: Γ at $t = T/2$

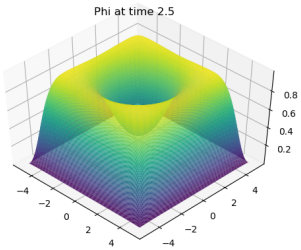

Figure 20: ϕ at $t = T/2$

However, we have to note that because of the time-consumption of the algorithm in 2d, we couldn't find the time to choose a discretization precision ensuring perfect mass and energy conservation, but as this seems to have close to no influence on the bulk of the distribution we can neglect it in first approximation.

4.2 First application to crowd modeling

In the work by Alexandre Nicolas and al, a cylinder is intruding into a uniformly distributed crowd and crossing it in a straight line. They study different situations (when people are asked not to anticipate the arrival of the cylinder, when the cylinder is coming from behind them,

etc.) but in our case, we will consider the agents always anticipate the cylinder movement (which is the whole point of mean field game theory). This corresponds in the experiment to the case where people face the cylinder and they see it arriving. In a first attempt to model the situation, we arbitrarily took the parameters to be $T = 1$, $L = 1$, $g = -2$, $dt = 0.005$, $dx = dy = \frac{1}{40}$, $\sigma = 0.45$, a cylinder radius of 0.25 and a potential wall strength of 100. The cylinder moves so that it starts from $(0, -1)$ at $t = 0$ and stops at $(0, 1)$ at $t = T$. In the meanwhile, it moves at constant velocity and in a straight line.

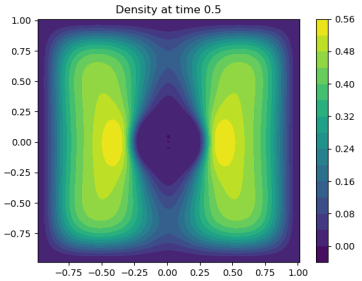

Figure 21: m at $t = T/2$

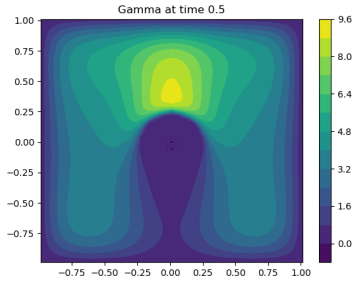

Figure 22: Γ at $t = T/2$

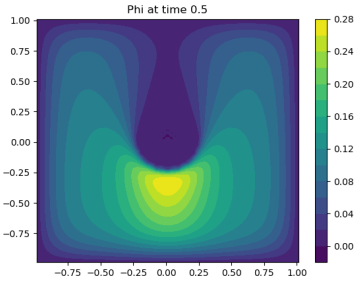

Figure 23: ϕ at $t = T/2$

Even though our mean field model is very simple, we can already see on fig 21-22 encouraging results. Indeed, the density is decreasing **before** and **after** the cylinder. This is what is described on fig 2 but here, we managed to qualitatively model it for the first time. Similarly, the "wings" observed in both figures shows these are due to long term anticipation that is taken into account by our mean field model. It shows that even for dense crowds, granular and fluids fails to describe accurately the situation because of their lack of anticipation. On the other hand fig 23 describes how the value function evolves. It is quite interesting to see that indeed, in front of the cylinder it is near 0 and that the closer people are to the trajectory of the center of the cylinder, the more they want to avoid it. This suggests that this model also takes into account the time needed to get out of the cylinder path. It is also logical that the backward equation is the one describing anticipation. On the other hand, Γ is high in front the cylinder, a place people are expected to find repulsive. This suggests that if ϕ describes the attractiveness of space, Γ describes its repulsiveness, and the final situation is a balance between the two. Finally, the wing structure of the solution do not appears explicitly on fig 22-23, which suggest that it may be a consequence of the backward-forward structure of the system, meaning a fully backward or fully forward system may not be able to model it. If it is the case, it means that it is the adaptive anticipation hypothesis that allows this model to find these qualitatively satisfying results. If people knew everything from start and perfectly anticipated from time 0 to the end what would happen, or if they were not to anticipate at all, these structures would not appear. It is henceforth a consequence of imperfect information.

5 Conclusion

During this internship, I had to understand how to apply a physicist mindset to a problem where it initially doesn't apply. In order to do that, I had to understand the link between "nonphysical" hypothesis (in the sense where they are not a consequence of a purely physical law) and the new formalism of mean-field games. Indeed, we wanted to used this formalism to model specific behaviors in crowd dynamics that was not explained by previous models. Hence, I was to implement an efficient code to solve a system of forward-backward equations. In this aspect, I had to confirm the efficiency and credibility of an already existing but not mathematically theorized 1d code by comparing it another one that was rigorously conceptualized. The first algorithm followed Crank-Nicholson algorithm with successive linear approximations of the system while the second consisted of a more specific numerical scheme combined with Newton method solving of the non-linear system. Special attention has been paid to initial, final conditions, and domain boundary conditions and their influence on mass and energy conservation. Curves obtained for m give sensible results while mass and energy are conserved up to discretization issue.
The code was then extended to 2d with relevant care, and its application to crowd behavior started. The first results obtained here suggested the model took into account several phenomena observed in experiment on crowd, namely anticipation and avoidance by pedestrian of a cylinder crossing a crowd, with its consequence the formation of a wing structure on the sides of the cylinder. It also suggested a specific role of ϕ and Γ and the fact the adaptive anticipation may be at the heart of the problem. More time would have been needed to construct definitive conclusions on the model, but what I already obtained suggests its relevance.
Indeed, although our results show very interesting qualitative behaviors in accordance with experiments, there is several points that still need to be elucidated. First, as soon as the simulation size increases, the time taken by the 2D code becomes very long. This suggests the use of parallelization to decrease the total time, and it has yet to be implemented.Tuning the parameters is necessary (essentially the healing length ν, which is equivalent to g) in order to obtain rigorous quantitative results to be compared with experiments. Lastly, the velocity in the hydrodynamic representation may be computed and related to real-world speed in order to compare with fig 3-4.

A Discretization used for numerical computation

Now just for completeness sake, we quickly extend the previous results to the discretization we used in our numerical computation.

$$\partial_x \Gamma = \frac{1}{2}\left(\frac{\phi_{i+1,j+1} - \phi_{i+1,j-1}}{2\Delta x} + \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta x}\right) \tag{32}$$

Which yields

$$\partial_{xx} \Gamma = \frac{1}{2\Delta x^2}\left[(\phi_{i+1,j+1} - 2\phi_{i+1,j} + \phi_{i+1,j-1}) + (\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1})\right] \tag{33}$$

and keeping the same discretization for time, we get

$$\frac{\phi_{i+1,j}^{n+\frac{1}{2}} - \phi_{i,j}^{n+\frac{1}{2}}}{\Delta t} + \frac{\sigma^2}{4\Delta x^2}\left[\left(\phi_{i+1,j+1}^{n+\frac{1}{2}} - 2\phi_{i+1,j}^{n+\frac{1}{2}} + \phi_{i+1,j-1}^{n+\frac{1}{2}}\right) + \left(\phi_{i,j+1}^{n+\frac{1}{2}} - 2\phi_{i,j}^{n+\frac{1}{2}} + \phi_{i,j-1}^{n+\frac{1}{2}}\right)\right] = -\frac{g}{2\sigma^2}\left((\phi_{i+1,j}^{n+\frac{1}{2}})^2 \Gamma_{i+1,j}^{n+1} + (\phi_{i,j}^{n+\frac{1}{2}})^2 \Gamma_{i,j}^{n+1}\right) \tag{34}$$

for ϕ and the following for Γ

$$\frac{\Gamma_{i+1,j}^{n+1} - \Gamma_{i,j}^{n+1}}{\Delta t} - \frac{\sigma^2}{4\Delta x^2}\left[\left(\Gamma_{i+1,j+1}^{n+1} - 2\Gamma_{i+1,j}^{n+1} + \Gamma_{i+1,j-1}^{n+1}\right) + \left(\Gamma_{i,j+1}^{n+1} - 2\Gamma_{i,j}^{n+1} + \Gamma_{i,j-1}^{n+1}\right)\right] = \frac{g}{2\sigma^2}\left(\phi_{i+1,j}^{n+\frac{1}{2}} (\Gamma_{i+1,j}^{n+1})^2 + \phi_{i,j}^{n+\frac{1}{2}} (\Gamma_{i,j}^{n+1})^2\right) \tag{35}$$

Introducing the same $\tilde{\sigma}$ and $\tilde{g}$ than previously, using the same notations we get an equation with spatial vectors

$$E_j(X_\bullet, Y_\bullet) = (Y_j - X_j) - \frac{\tilde{\sigma}}{2}[(Y_{j+1} - 2Y_j + Y_{j-1}) + (X_{j+1} - 2X_j + X_{j-1})] - \frac{\tilde{g}}{2}\left(\phi_{i+1,j}^{n+\frac{1}{2}} (Y_j)^2 + \phi_{i,j}^{n+\frac{1}{2}} (X_j)^2\right) \tag{36}$$

Where the last term is the two-time regularization of the potential. This can be solved by Newton algorithm in the exact same way as previously, except for a source term that is more complex.

Thanks

I would like to thank my supervisors and Thibault Bonnemain for their help during this intership, and all my friends you regularly played with me to make me forget I was passing my confinement alone far from both family and friends.

References

[1] Nicolas Alexandre et al. "Mechanical response of dense pedestrian crowds to the crossing of intruders". In: *Nature, Scientific Reports* (Jan. 2019).

[2] Thibault Bonnemain. "Quadratic mean field games with negative coordination". PhD thesis. Université Cergy-Pontoise, 2020.

[3] Jur van den Berg, C. Lin Ming, and Manocha Dinesh. *Reciprocal velocity obstacles for real-time multi-agent navigation*. Pasadena Calif. May 19-23 2008. 2008, pp. 1928–1935.

[4] Kolb, Cixous, and Charmet. "Flow fields around an intruder immersed in a 2d dense granular layer". In: *Granul. Matter* ***16*** (2014), pp. 223–233.

[5] Richard Bellman and Rand Corporation. *Dynamic Programming*. Princeton University Press, 1957.

[6] C.W. Cardinger. *Handbook of stochastic method for physics, chemistry, and the natural sciences*. Springer-Verlag, 1985.

[7] Jean-Michel Lasry and Pierre-Louis Lions. "Jeux à champs moyen. I - Le cas stationnaire". In: *Comptes Rendus Mathématique* (2006), pp. 619–625.

[8] Jean-Michel Lasry and Pierre-Louis Lions. "Mean field games". In: *Japanese Journal of Mathematics* (Mar. 2007), pp. 229–260.

[9] Denis Ullmo, Igor Swiecicki, and Thierry Gobron. "Quadratic Mean Field Games". In: *Physics Reports 799* (2019), pp. 1–35.

[10] Olivier Guéant. "New Numerical Methods for mean fields games with quadratic costs". In: *Network and heterogeneous media* (June 2012), pp. 315–336.

[11] Choy Yaan Yee et al. "Crank-Nicholson implicit method for the nonlinear Schrödinger equation with variable coefficient". In: *AIP Conference Proceedings 1605.1* (2014), pp. 76–82.

[12] *Eigen library documentation*. eigen.tuxfamily.org. June 2020.

[13] Pierre Cardaliaguet et al. "Long time average of mean field games with a nonlocal coupling". In: *SIAM J. Control Optim 51.5* (2013), pp. 3558–3591.